

**Chapter:**                    **7.0 Construction Stage**

**Description:**            The goal of this stage is to translate the set of technical, computer-oriented system design specifications into a language the computer can understand and execute. Construction involves coding, validation and unit testing by a developer. Any hardware or software procured to support the construction effort is installed. Plans are developed for the installation of the operating environment hardware and software. A training program is designed and a Training Plan that describes the system is produced.

The activities in this stage result in the transformation of the system design into the first complete executable representation of the product. If required, the source code or COTS “glue” code, including suitable comments, is generated using the approved program specifications. The source code is then grouped into processable units and all high-level language units are compiled into object code. Unit testing is performed to determine if the code satisfies the program specifications and is complete, logical, and error free.

The operating documentation is also produced. The operating documentation is required for installing, operating, and supporting the product through its lifecycle.

**Input:**                      The following items provide input to this stage:

SEM Templates:

- Conversion Plan
- Functional Design Document
- Maintenance Plan
- Requirements Specification
- Requirements Traceability Matrix
- System Design Document
- Test Plan
- Test Reports

PMM Templates:

- Project Plan
- Quality Management Plan
- Security Plan

Other Inputs:

- Data Dictionary

**High-Level Activities:**

The remainder of this chapter is divided into sections that describe the specific high-level activities performed during this stage. These activities represent the minimum requirements for a large information systems engineering effort. Notes are provided, as applicable, to assist in customizing these lifecycle stage requirements and to accommodate the different sizes of information systems engineering efforts. The high-level activities are presented in the sections listed below.

- 7.1 Establish Development Environment
- 7.2 Develop Programs
- 7.3 Conduct Unit Testing
- 7.4 Establish Development Baselines
- 7.5 Plan Transition to Operational Status
- 7.6 Generate Operating Documentation
- 7.7 Develop Training Plan
- 7.8 Develop Installation Plan

**Touch Points:**

The following touch points are involved in the Construction Stage:

- Contracts and Procurement
  - Contract Liaison involvement if contract issues arise
- E-Michigan
  - Continue to work with E-Michigan's webmaster, as appropriate, to ensure ADA compliance and Michigan.gov look and feel standards
- Infrastructure Services
  - Infrastructure Specialist involvement in establishing hosting environments
- Security
  - Finalize Network and Data Flow diagrams

**Output:**

Several work products are produced during this stage. The work products listed below are the minimum requirements for a large project. Deviations in the content and delivery of these work products are determined by the size and complexity of the project. Explanations of the work products are provided under the applicable activities described in the remainder of this chapter.

**SEM Templates:**

- Conversion Plan (*revised*)
- Installation Plan (*initial*)
- Maintenance Plan (*revised*)
- Requirements Traceability Matrix (*revised*)
- Test Plan (*final*)

- Test Reports (*revised*)
- Training Checklist (*final*)
- Training Plan (*initial*)
- Transition Plan (*initial*)

PMM Templates:

- Project Plan (*revised*)
- Security Plan (*revised*)

Other Outputs:

- Development baselines
- Operating Documentation
  - Users Manual
  - Developer's Reference Manual
- Project Test File
- System units and modules

A diagram showing the work products associated with each high-level activity is provided in *Exhibit 7.0-1, SEM Overview Diagram – Construction Stage Highlighted*.

Review the Project Plan for accuracy and completeness of all Construction Stage activities and make any changes needed to update the information.

**Review Process:**

Quality reviews are necessary during this stage to validate the product and associated work products. The activities that are appropriate for quality reviews are identified in this chapter and Chapter 2.0, Lifecycle Model. The time and resources needed to conduct the quality reviews should be reflected in the project resources, schedule, and work breakdown structure.

**Structured Walkthrough (SWT)**

Requirements for a peer review or a more formal structured walkthrough are documented under *Review Process* at the end of each Task, Subtask, or Activity section in this stage. The State of Michigan guide titled *Structured Walkthrough Process Guide* provides a procedure and sample forms that can be used for SWTs. This document is available on the MDIT SUITE website.

**Stage Exit**

Schedule a Stage Exit as the last activity of the Construction Stage to enable the project approvers to review project deliverables and provide a concur/non-concur position to the project manager. The State of Michigan guide titled *Stage Exit Process Guide* provides a procedure and sample report form that can be used for stage exits. This document is available on the MDIT SUITE website.

**References:**

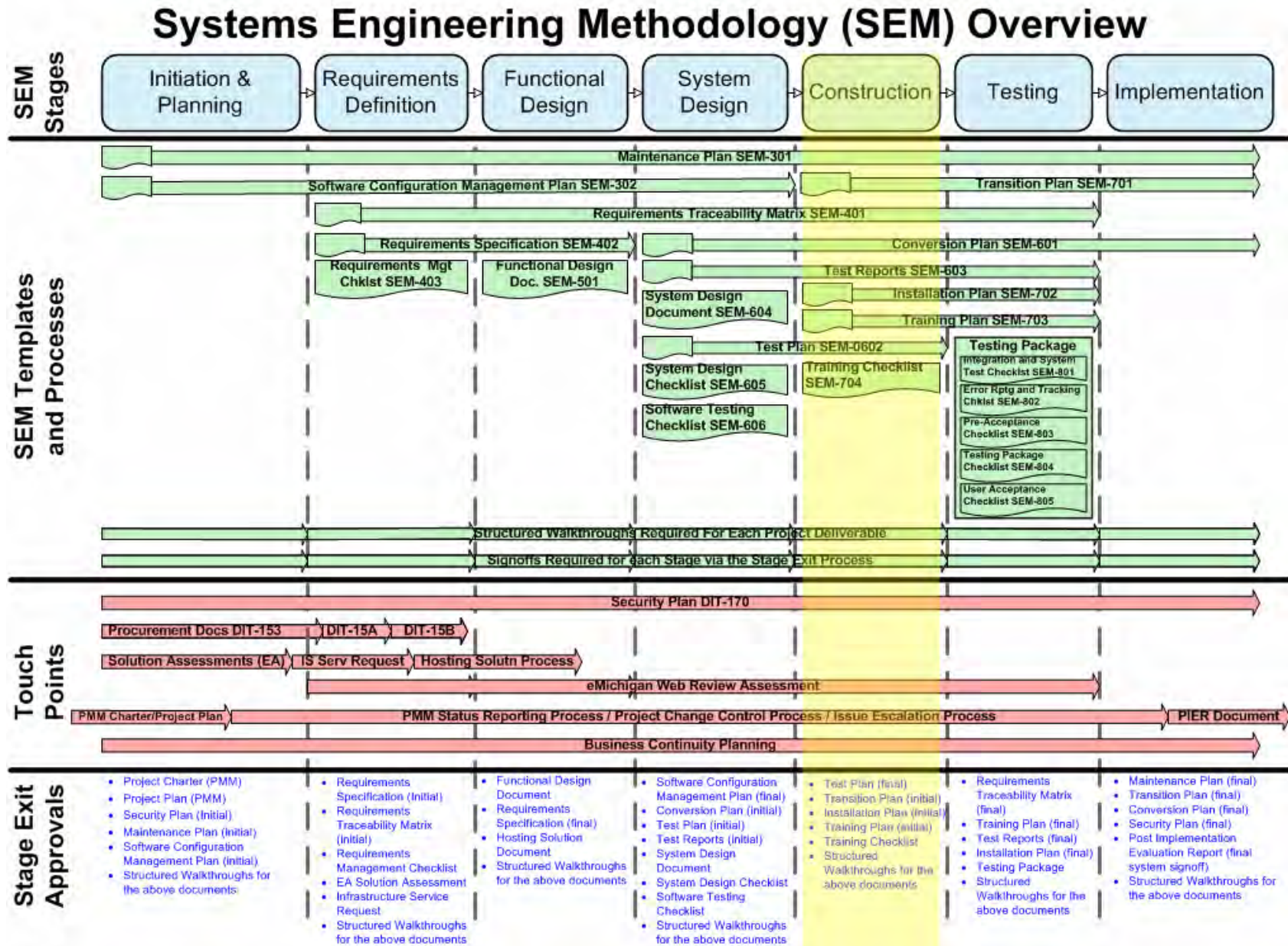
Chapter 2.0, Lifecycle Model, *Quality Reviews* provides an overview of the Quality Reviews to be conducted on a project.

**Bibliography:**

The following materials were used in the preparation of the Construction Stage chapter.

1. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard for Developing Software Lifecycle Processes*, IEEE Std 1074-1991, New York, 1992.
2. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard for Software User Documentation*, IEEE Std 1063-2001, New York, 2001.
3. U.S. Department of Energy, *Automated Data Processing Systems Development Methodology, Volume 1*, K/CSD/INF/86-3, Vol.1, R3, prepared under contract by Martin Marietta Energy Systems, Inc. at Oak Ridge National Laboratory, August 1987.
4. U.S. Department of Energy, *DOE/NV Software Management Plan*, Nevada Operations Office, May 1991.
5. U.S. Department of Justice, Immigration and Naturalization Service, *System Development Lifecycle Standards Manual*, 1991.
6. The Institute of Electrical and Electronics Engineers, Inc., *Guide to the Software Engineering Body of Knowledge, Stone Man Trial Version 1.00*, New York, May 2001.
7. Carnegie Mellon University, Software Engineering Institute, *Capability Maturity Model: Guidelines for Improving the Software Process*, Addison Wesley Longman, Inc., 1994.
8. Microsoft Press, *Code Complete, Volume 2*, Michael McConnell, 2004, ISBN: 0-7356-1967-0.

Exhibit 7.0-1 SEM Overview Diagram – Construction Stage Highlighted



**Activity:** 7.1 Establish Development Environment

**Responsibility:** Project Team

**Description:** Establishing the development environment involves assembling and installing the hardware, software, communications equipment, databases, and other items required to support the coding/construction effort. When the installation of the equipment or software by MDIT Technical Services staff is complete, conduct testing to verify the operating characteristics and functionality of the hardware and software. If required, security software and procedures should be activated when the installations are completed.

If the operational environment is also the development environment, it may be necessary to alter the operational environment to accommodate an infrastructure of purchased hardware and software for use during construction and testing.

Before being integrated into or used to support the product, vendor products should be tested to verify that the product satisfies the following objectives.

- The product performs as advertised/specified.
- The product's performance is acceptable and predictable in the target environment (e.g., testing for LAN certification).
- The product fully or partially satisfies the project requirements.
- The product is compatible with the project team's other hardware and software tools.

Time should be planned for the project team to become familiar with new products. Ensure that the project team members who will use the hardware or software obtain proper training. This may involve attendance at formal training sessions conducted by the vendor or the services of a consultant to provide in-house training.

This is a good time to review the coding practices that were established in the System Design Stage. Make any changes to the standards that are needed to accommodate the procured hardware and software.

**Activity:** 7.2 Develop Programs

**Responsibility:** Project Team Developers

**Description:** This activity involves generating the source and object code for the product. The code should be written in accordance with the coding practices developed in the System Design Stage. The use of various development languages will dictate whether the code is manually prepared or automatically generated. Regardless of the platform, construction of the code should adhere to a consistent set of coding practices and error prevention procedures. This will promote reliable, maintainable code, developed in the most efficient and cost effective manner.

Code units should be generated in a sequence based on a plan that accounts for factors such as criticality, difficulty, integration and test issues, and needs of customers and users, as appropriate.

For COTS products, although software COTS products attempt to simulate the "plug and play" capability of the hardware world, frequently this is not the case. Most products require some amount of adaptation and integration to work within the State of Michigan environment. The typical solution is to adapt each software COTS product through the use of "wrappers," "bridges," or other "glueware." It is important to note that adaptation does not imply modification of the COTS product. Adaptation can be a complex activity that requires technical expertise at the detailed system and specific COTS component levels. Adaptation and integration must take into account the interactions among custom components, COTS products, any non-developmental item components, any legacy code, and the architecture including infrastructure and middleware elements.

The source and object code should be uniquely identified and stored in a way to facilitate the configuration control measures described in the Software Configuration Management Plan.

Developing programs includes the following tasks.

- Use the Program Specifications developed in the System Design Stage as the basis for the coding effort.
- Generate source code.
- Generate the physical files and database structure.
- Generate video screens and report generation codes.
- Configure predefined options for COTS products.

If conversion of an existing system or data is necessary, generate the program(s) described in the Conversion Plan.

- Conduct preliminary unit testing. When the test output is correct, review the program specification to assure that the unit or module conforms to the specification.

**Coding Practices:** The following coding practices should be implemented.

- The code development staff should meet at scheduled intervals to discuss problems encountered and to facilitate program integration and uniformity.
- Program uniformity should be achieved by using a standardized set of naming conventions for programs, data elements, variables, and files.
- Modules that can be shared by programs requiring the same functionality should be implemented to facilitate development and maintenance.
- Modules that can be borrowed or reused from other sources that have already been tested should be implemented. Code reuse can lead to faster development.
- Meaningful internal documentation should be included in each program.
- All code should be backed up on a daily basis and stored in an offsite location to avoid catastrophic loss.
- A standard format for storing and reporting elements representing numeric data, dates, times, and information shared by programs should be determined.
- The System Design Document should be updated to reflect any required deviations from the documented design.

**Work Products:** The following work products are produced.

- Completed units and modules of code.
- Configured COTS options (e.g., databases, tables)
- Test materials generated from preliminary testing.

Each requirement identified in the Requirements Specification must be traceable to the code. This traceability ensures that the product will satisfy all of the requirements and will not include inappropriate or extraneous functionality.



Expand the Requirements Traceability Matrix developed in the Requirements Definition Stage to relate the source and object code to the requirements. Place a copy of the expanded matrix in the Project File.

**Review Process:** Periodic informal reviews of each developer's work are recommended to keep the project team informed of progress and to facilitate the resolution of any problems that may occur. The combined knowledge and skills of the team members will help to build quality into the product and support the early detection of errors in design, logic, or code.

Conduct structured walkthroughs on the expanded Requirements Traceability Matrix and completed units and modules to assure that the code is accurate, logical, internally well documented, complete, and error free. Structured walkthroughs should also be used to validate that the code is reliable and satisfies the program specifications and project requirements.

For large or complex projects, code inspections may be a more effective method of removing defects and identifying areas where defects may be propagated. Conduct the code inspections at successive stages of code production. Code inspection is a static analysis technique that relies on visual examination of code to detect errors, violations of development standards, and other problems. These inspections are particularly important when code is being produced by several developers or different teams. The inspection team may include experts outside of the project. Ideal times for code inspections occur when code and unit tests are complete, and when the first integration tests are complete. Code inspections should be identified as milestones in the Project Plan.

**Activity:** 7.3 Conduct Unit Testing

**Responsibility:** Project Team Developers

**Description:** Unit testing is used to verify the input and output for each module. Successful testing indicates the validity of the function or sub-function performed by the module and shows traceability to the design. During unit testing, each module is tested individually and the module interface is verified for consistency with the design specification. All important processing paths through the module are tested for expected results. All error handling paths are also tested.

Unit testing is driven by test cases and test data that are designed to verify requirements, and to exercise all program functions, edits, in-bound and out-bound values, and error conditions identified in the program specifications. If timing is an important characteristic of the module, tests should be generated that measure time critical paths in average and worst-case situations.

Plan and document the inputs and expected outputs for all test cases in advance of the tests. Log all test results. Analyze and correct all errors and retest the unit using the scenarios defined in the test cases. Repeat testing until all errors have been corrected.

While unit testing is generally considered the responsibility of the developer, the project manager or lead developer should be aware of the unit test results.

**Work Products:** Completion of unit testing for a component signifies internal project delivery of a component or module for integration with other components. Place all components that have completed unit testing under configuration control as described in the Software Configuration Management Plan. These components form the Production Baseline. Configuration controls restrict changes to tested and approved software in the Production Baseline. Subsequent changes or additions to the software that are agreed upon in a System Design Review and receive stakeholder concurrence supersede the existing baseline and establish a new Production Baseline.

Review the draft versions of the Test Plan developed during the System Design Stage. Update the plan, as needed, to reflect any changes made to the design. Deliver the final versions of the Test Plan to the system owner and user for review and approval. Place a copy of the approved plan in the Project File.

Create a Project Test File for all test materials generated throughout the project lifecycle. Place all unit test materials (e.g., inputs, outputs, results and error logs) in the Project Test File. The test cases used for unit testing may become a subset of tests for integration testing.

***Review Process:*** Conduct peer reviews on the test materials to be placed in the Project Test File. Conduct structured walkthroughs on any updated plans (e.g., Integration and System Test Requirements).

**Activity:** 7.4 Establish Development Baselines

**Responsibility:** Project Team Developers

**Description:** A development baseline is an approved "build" of the product. A build can be a single component or a combination of components. The first development baseline is established after the first build is completed, tested, and approved by the project manager or lead developer. Subsequent versions of a development baseline should also be approved. The approved development baseline for one build supersedes that for its predecessor build.

Conduct internal build tests such as regression, functional, performance, and reliability. Regression tests are designed to verify that capabilities in earlier builds continue to work correctly in subsequent builds. Functional tests focus on verifying that the build meets its functional and data requirements and correctly generates each expected display and report. Performance and reliability tests are used to identify the performance and reliability thresholds of each build.

Once the first development baseline is established, any changes to the baseline must be managed under the change control procedures described in the Software Configuration Management Plan. Approved changes to a development baseline must be incorporated into the next build of the product and revisions made to the affected work products (e.g., Requirements Specification, System Design Document, and Program Specifications).

**Work Product:** Document the internal build test procedures and results. Identify errors and describe the corrective action that was taken. Place a copy of the internal build test materials in the Project Test File.

Maintain configuration control logs and records as required in the Software Configuration Management Plan.

Expand the Requirements Traceability Matrix developed in the Requirements Definition Stage. All work products developed during the code, unit testing, and build processes must be traced back to the project requirements and system design. This traceability ensures that the product will satisfy all of the requirements and remain within the project scope. Place a copy of the expanded Requirements Traceability Matrix in the Project File.

**Review Process:** Conduct peer reviews on the internal build test materials to be placed in the Project Test File. Conduct structured walkthroughs on any updated documents (e.g., the Requirements Traceability Matrix).

<b>Activity:</b>	<b>7.5 Plan Transition to Operational Status</b>
<b>Responsibility:</b>	Project Team
<b>Description:</b>	<p>Successful transition from acceptance testing to full operational use of the product depends on planning the transition long before the product is installed in its operational environment. In planning for the transition, quantify the operational needs associated with the product and describe the procedures that will be used to perform the transition. Rely on experience and data gathered from previous, similar projects to define these needs.</p>
<b>Work Product:</b>	<p>Develop a Transition Plan that describes the detailed plans, procedures, and schedules that will guide the transition process. Coordinate development of the plan with the operational and maintenance personnel. The following issues should be considered in the preparation of a Transition Plan.</p> <ul style="list-style-type: none"><li>• Develop detailed operational scenarios to describe the functions to be performed by the operational support staff, maintenance staff, and users.</li><li>• Document the release process. If development is incremental, define the particular process, schedule, and acceptance criteria for each release.</li><li>• Describe the development or migration of data, including the transfer or reconstruction of historic data. Schedule ample time for the system owner and user to review the content of reconstructed or migrated data files to reduce the chance of errors or omissions.</li><li>• Specify problem identification and resolution procedures for the operational product.</li><li>• Define the configuration management procedures that will be used for the operational product. Ideally, the methods defined in the Software Configuration Management Plan that were employed during product development can continue to be used for the operational product.</li><li>• Define the scope and nature of support that will be provided by the project team during the transition period.</li><li>• Specify the organizations and individuals who will be responsible for each transition activity, ensuring that responsibility for the product by the operations and maintenance personnel increases progressively.</li><li>• Identify products and support services that will be needed for day-to-day operations or that will enhance operational effectiveness.</li></ul>

***Review Process:*** Conduct a structured walkthrough to assure that the Transition Plan is logical, accurate, and complete. Involve operational and maintenance personnel in the walkthrough.

***Resource:*** A Transition Plan is available on the MDIT SUITE website.

**Activity:** 7.6 Generate Operating Documentation

**Responsibility:** Project Team/Technical Writer

**Description:** Plan, organize, and write the operating documentation that describes the functions and features of the product from the users point-of-view. The different ways that users (including system administration and maintenance personnel) will interact with the product must be considered. The needs of the users should dictate the document presentation style and level of detail. Responsibilities for changing and maintaining the documents should be described in each document.

The following are typical operating documents for a large project.

- Users Manual/Online Help Screens
- Developer's Reference Manual
- Systems Administration Manual
- Database Administration Manual
- Operations Manual

It is recommended that a technical writer be involved in the generation of all operating documents. A technical writer works closely with the project team to ensure that documents are grammatically correct; comply with applicable standards; and are consistent, readable, and logical.

**Note:** The operating documents can be produced as separate manuals or combined to accommodate less complex projects.

**Procedure:** Use the following procedure to develop the operating documentation.

- Identify the operating documents that need to be developed. Determine if any of the documents can be combined or delivered as multiple volumes.
- Determine whether the documents should be provided as printed material, standalone electronic files, online documentation accessed through the product, or a combination.
- Determine the best presentation method or combination of methods required for each of the documents, such as a traditional manual, quick reference guide or card, or online help.
- Identify all of the features of the user interface and the tasks users will perform.
- Identify the users' needs and experience levels to determine:

- The amount of user interaction, level of interaction, and whether the interaction is direct or indirect.
  - The appropriate level of detail (e.g., the Users Manual should not contain highly technical terms and explanations that may confuse or frustrate a user).
- Determine the document content and organization based on whether the document will be used more as an instructional tool or a reference guide.
- Develop descriptions of each function and feature of the product and organize the information to facilitate quick, random access.
- Provide appropriate illustrations and examples to enhance clarity and understanding.
- Establish a schedule for the documents to be reviewed after the product goes into production. Operating documents must be kept up-to-date as long as the product remains in production.

**Work Products:** Refer to each of the following tasks for applicable work products.

**Review Process:** Refer to each of the following tasks for applicable review processes.

**Tasks:** The following tasks describe the minimum requirements for operating documentation.

7.6.1 Produce Users Manual

7.6.2 Produce Developer's Reference Manual



**Task:** 7.6.1 Produce Users Manual

**Description:** The Users Manual provides detailed information users need to access, navigate through, and operate the product. Users rely on the Users Manual to learn about the product or to refresh their memory about specific functions. A Users Manual that is organized functionally so that the information is presented the same way the product works helps users understand the flow of menus and options to reach the desired functions.

Different categories of users may require different types of information. A modular approach to developing the Users Manual to accommodate the needs of different types of users eliminates duplication and minimizes the potential for error or omission during an amendment or update. For example, separate general information that applies to all users from the special information that applies to selected users such as system administrators or database administrators. The special information can be presented in appendixes or supplements that are only provided to the users who need the information.

**Work Product:** Write the draft Users Manual in clear, non-technical terminology that is oriented to the experience levels and needs of the user(s). The following are typical features of a users manual.

- Overview information on the history and background of the project and the architecture, operating environment, and current version or release of the product.
- Instructions for how to install, setup, or access the product.
- Complete coverage of all functions, presented in a logical, hierarchical order.
- Accurate pictures of screens and reports, ideally with data values shown, so the user can easily relate to examples.
- In-depth examples and explanations of the areas of the product that are most difficult to understand.
- Clear delineation of which features are accessible only to specific users.
- Instructions on accessing and using online help features.
- Procedures for data entry.

- Descriptions of error conditions, explanations of error messages, and instructions for correcting problems and returning to the function being performed when the error occurred.
- Instructions for performing queries and generating reports.
- Who to contact for help or further information.

**Note:**

For large or complex products, separate manuals (e.g., User's Manual, Database Administrator's Manual, and System Administrator's Manual) may be necessary to address the needs of different categories of users.

For very small projects, a quick reference guide or card may be more appropriate than a full-scale Users Manual. The guide or card should be designed to provide a quick reference of logon, logoff, and commands for frequently used functions.

For projects of any size, a quick reference card may be developed as a supplement to more detailed user documentation.

**Review Process:**

Conduct structured walkthroughs for the draft Users Manual or set of user documents to assure that the documentation is complete, easy to use, and accurately reflects the product and its functions.

The draft user documentation will be tested and verified with the product during the Testing Stage.

**Task:** 7.6.2 Produce Developer's Reference Manual

**Description:** The Developer's Reference Manual contains information about program development used by the maintenance staff to maintain the programs, databases, interfaces, and operating environment. The Developer's Reference Manual should provide an overall conceptual understanding of how the product is constructed and the details necessary to implement corrections, changes, or enhancements.

The Developer's Reference Manual describes the logic used in developing the product and the functional and system flow to help the maintenance staff understand how the programs fit together. The information should enable a developer to determine which programs may need to be modified to change a system function or to fix an error.

**Work Product:** The following are typical features of a Developer's Reference Manual.

- A description of the technical environment, including versions of the development language(s) and other proprietary software packages.
- A brief description of the design features including descriptions of unusual conditions and constraints.
- An overview of the architecture, program structure, and program calling hierarchy.
- The design and coding practices and techniques used to develop the product.
- Concise descriptions of the purpose and approach used for each program.
- Layouts for all data structures and files used in the product.
- Descriptions of maintenance procedures, including configuration management, program checkout, and system build routines.
- The instructions necessary to compile, link, edit, and execute all programs.
- Manual and automated backup procedures.
- Error-processing features.

Use appendixes to provide detailed information that is likely to change as the product is maintained. For example, a list of program names and a synopsis of each program could be included as an appendix.

***Review Process:*** Conduct structured walkthroughs of the draft Developer's Reference Manual to assure that the documentation is complete, easy to use, and accurately reflects the product and its functions.

The draft Developer's Reference Manual will be tested and verified with the product during the Testing Stage.

**Activity:** 7.7 Develop Training Plan

**Responsibility:** Project Team

**Description:** A Training Plan defines the training needed to implement and operate the product successfully. The Training Plan should address the training that will be provided to the system owner, users, and maintenance staff. When new hardware or software is being used, affected personnel will need hands-on experience before bringing the new system (equipment and/or software) into daily operation.

Training must address both the knowledge and the skills required to operate and use the system effectively. Design the training to accomplish the following objectives.

- Provide trainees with the specific knowledge and skills necessary to perform their work.
- Prepare training materials that will sell the product as well as instruct the trainees. The training should leave the trainees with the enthusiasm and desire to use the new product.
- Account for the knowledge and skills the trainees bring with them, and use this information as a transition to learning new material.
- Anticipate the needs for follow-on training after the product is fully operational, including refresher courses, advanced training, and repeats of basic courses for new personnel.
- Build in the capability to update the training as the product evolves.

Involve the system owner and key users in the planning to determine the education and training needs for all categories of users (managers, users, and maintenance staff).

The Training Plan should address the following issues:

- Identify the organization's training policy for meeting training needs.
- Ensure software managers have received orientation on the training.
- Ensure training courses prepared at the organization level are developed and maintained according to organizational standards.

- Ensure a procedure for required training is established and used to determine whether individuals already possess the knowledge and skills required to perform in their designated area.
- Ensure measurements are made and used to determine the status of training activities.
- Ensure that training activities are reviewed with senior management on a periodic basis.
- Ensure the training is independently evaluated on a periodic basis for consistency with, and relevance to, the organization's needs.
- Ensure the training activities and work products are reviewed and/or audited and the results are reported.
- Ensure training records are properly maintained.

**Work Product:**

Prepare a draft Training Plan that describes the training and at a minimum addresses the following issues.

- Identifies personnel to be trained. Review the list of trainees with the system owner and users to ensure that all personnel who should receive training have been identified.
- Defines the overall approach to training and the required training courses.
- Establishes the scope of the training needed for users, management, operations, and maintenance personnel.
- Defines how and when training will be conducted. Specify instructor qualifications, learning objectives, and mastery or certification requirements (if applicable).
- Identifies any skill areas for which certification is necessary or desirable. Tailor the training to the certification requirements.
- Establishes a preliminary schedule for the training courses. The schedule must reflect training requirements and constraints outside the project. Schedule individual courses to accommodate personnel who may require training in more than one area. Identify critical paths in the training schedule such as the time period for the product's installation and conversion to production status.

- Defines the required course(s), outlines their content and sequence, and establishes training milestones to meet transition schedules.
- Tailors the instruction methods to the type of material being presented. Include classroom presentation, interactive computer-assisted instruction, demonstrations, individual video presentations, and hands-on experience, either live or simulated.
- Identifies trainers who are technically knowledgeable and were involved in the design and development of the system. For projects with extensive and formal training requirements, it may be necessary to provide training for the trainers.
- Consider availability of the following: users, system-tested software, training rooms and equipment, and the completion of system documentation and training materials.

Complete the Training Checklist to ensure that all activities and work products are complete.

Place a copy of the initial Training Plan and completed Training Checklist in the Project File. The plan will be reviewed and updated during the Testing Stage.

***Review Process:*** Conduct a structured walkthrough to assure that the initial Training Plan is accurate and complete.

***Resource:*** The Training Plan template and Training Checklist are available on the MDIT SUITE website.

<b>Activity:</b>	<b>7.8 Develop Installation Plan</b>
<b>Responsibility:</b>	Project Team
<b>Description:</b>	<p>The Installation Plan is prepared to specify the requirements and procedures for the full-scale installation of the developed product at the system owner's and all users' work sites. The plan also addresses the installation of any hardware, off-the-shelf software, firmware, and communications equipment needed to operate the product at each site. In developing an Installation Plan consider each site's requirements for continuity of operations, level of service, and the needs of the project team, users, maintenance personnel, and management.</p>
<b>Work Product:</b>	<p>Work closely with the system owner and representatives from the user sites to assure that all site-specific hardware, software, and communications installation requirements are addressed in the Installation Plan. Develop an initial Installation Plan that addresses the following issues.</p> <ul style="list-style-type: none"><li>• Schedule of all installation activities.</li><li>• Items to be delivered to each installation site.</li><li>• Number and qualifications of personnel performing installation.</li><li>• Equipment environmental needs and installation instructions.</li><li>• Hardware, software, firmware, tools, documentation, and space required for each installation.</li><li>• Special requirements governing the movement of equipment to each site.</li><li>• Communications requirements.</li><li>• Dependencies among activities affected by installation.</li><li>• Installation tests to assure the integrity and quality of the installed product.</li></ul> <p>Ensure any special requirements regarding, e.g., network resources and web-based applications are adequately documented. Place a copy of the initial Installation Plan in the Project File.</p>
<b>Review Process:</b>	<p>Conduct a structured walkthrough to assure that the initial Installation Plan is accurate and complete. The Installation Plan will be reviewed and revised as needed during the Testing and Implementation Stages.</p>
<b>Resource:</b>	<p>An Installation Plan template is available on the MDIT SUITE website.</p>